APPLICATION FOR UNITED STATES LETTERS PATENT

For

Method and Apparatus For Efficient Decimation Based Correlation Technique For Identifying a Looked For Word

Inventor:

Darren Bowler

Prepared by:

BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP
12400 Wilshire Boulevard
Los Angeles, CA 90025-1030
(408) 720-8300

Attorney's Docket No.: 04148.P014

Method and Apparatus For Efficient Decimation Based Correlation Technique For Identifying a Looked For Word

PRIORITY

[0001] This application hereby claims the benefit of a U.S. Provisional Application filed on September 1, 2000 and provided Application No. 60/230,167.

Field of Invention

[0002] The field of invention relates generally to signal processing; and, more specifically, to a method and apparatus for an efficient decimation based correlation technique for identifying a looked for word.

Background

[0003] **Figure 1** shows an example of a BLUETOOTH packet structure 120. The packet may be viewed as having two parts: 1) an access code 125; and 2) the remainder of the packet 160. The remainder of the packet 160 typically includes a packet header and the random "customer" data the packet is responsible for transporting. The access code 125 is unique to the piconet for the connection between the transmitting device 165 and the receiving device 166. In BLUETOOTH applications, typically, the access code 125 includes a 4 bit preamble 121, a 64 bit synchronization word 122, and a 4 bit postamble 123.

[0004] The synchronization word 122 (which may also be referred to as a synch word, synchronization code, synch code, and the like) is unique to the

piconet used in the connection. Upon the reception of a packet at the receiving

device 166, the receiving device 166 "checks" the synchronization word 122

embedded within the packet.   If the synchronization word 122 matches a

pseudo random sequence unique to the piconet master for a connection that

includes the receiving device 166, the receiving device 166 understands that the

incoming packet is intended for the receiving device 166.

[0005]   **Figure 2** shows an exemplary embodiment of some of the functional

blocks within the receiving device that may be used to perform the above

described synchronization word 122 check. **Figure 2** shows a receive channel

209 having a demodulator 206 followed by a plurality of functional components.

For BLUETOOTH applications, demodulator 206 corresponds to a frequency

shift keyed (FSK) demodulator (which may be implemented as a frequency to

voltage converter). An analog to digital converter 201 receives the demodulator

206 output signal 255.

[0006]   The demodulator 206 output signal 255 may also be referred to as a

baseband signal 255. The baseband signal 255 may be filtered, amplified (or

processed in other ways) between the demodulator 206 output and the A/D

converter 201 input. For purposes of discussing exemplary embodiments, the

baseband signal 255 (as it exists prior to processing by the A/D converter 201)

may be viewed as an analog signal.

[0007]   The analog to digital converter 201 is responsible for converting the analog baseband signal 255 into a series of words having values representative of the analog baseband signal waveform. A word is a plurality of bits (where the number of bits may be given generically as "n"). An A/D converter 201 output word may also be referred to as a sample, an output sample, an output word sample, and the like.

[0008]   The A/D converter output signal 203 is provided to a slicer unit 210. The slicer unit 210 converts the A/D output signal 203 into samples of recovered symbols. That is, note that the pulses 211a,b, 212a,b, 213a,b observed in both the baseband signal 255 and the A/D converter output signal 203 are representative of 1s or 0s being transmitted from the transmitting device to the receiving device.

[0009]   The slicer unit 210 effectively identifies the presence of each pulse 211a,b, 212a,b, 213a,b and reports the binary value (i.e., a "1" or a "0") of each A/D converter output sample to the correlator 221 (as seen in the slicer output signal 222 of **Figure 2**). For example, the slicer unit 210 may be designed to make a determination of the average value of the A/D converter output signal 203 and threshold the individual values of the A/D converter output signal 203 against this average value.

4

[0010]   **Figure 2** shows an example. The average value of the A/D converter

output signal 203 is seen at word value level 217. The slicer unit 210 determines

this level 217 based upon the observed data stream from the A/D converter 201

(e.g., by continually averaging its output values). The slicer unit 210 converts

A/D output values into "1"s or "0"s based upon their position with respect to the

threshold level 217.

[0011]   That is, A/D converter output signal 203 values above the threshold

level 217 are given a value of "1" by the slicer unit 210 while A/D converter output

signal 203 values below the threshold value 217 are given a value of "0" by the

slicer unit 210.   The activity of deciding whether a signal is a "1" or a "0" based

upon its level with respect to a reference (such as threshold 217 mentioned

above) may be referred to as slicing, thresholding, comparing and the like. The

slicer unit output signal 222 is shown in **Figure 2**. Note that it traces a digital

symbol signal 224 which is shown in **Figure 2** for conceptual ease. Digital

symbol signal 224 does not need to actually exist at the slicer 210 output.

[0012]   The correlator unit 221 performs a correlation between the received

synchronization word and the synchronization word that the receiving device is

"looking for". A correlation employs a mathematical process (e.g., a

"convolution") which may be implemented with electronic circuitry or software.

That is, the correlator unit 221 may constructed with logic that performs a

correlation or may be implemented as a processor that executes a correlation

software routine. The correlator unit provides a measurement of the likeness between two signals.

[0013] To perform the aforementioned synchronization word check, the received synchronization word from the slicer unit 210 is correlated with the synchronization word that is unique to the piconet master of the receiving device's connection. Because the synchronization word that is unique to the piconet master of the receiving device's connection is that basis against which the correlation is performed, it may be referred to as "the correlation word", "the looked for synchronization word", "the looked for word" and the like. The correlation word 230 is drawn in Figure 2 as an input to the correlator 221.

[0014] Thus, the correlator 221 "checks" the likeness between the correlation word 230 and the slicer output signal 222. Eventually, the slicer output signal 222 presents the synchronization word 122 of the received packet. If they are deemed to have an acceptable amount of likeness, the synchronization words are deemed to be the same and the received packet is regarded as being intended for the receiving device.

[0015] A problem, however, with using correlation as a vehicle for identifying a "looked for" synchronization word is poor efficiency (e.g., in the form of large numbers of A/D converter 201 output samples needed to properly execute the correlation process).

BRIEF DESCRIPTION OF THE DRAWINGS

[0016]   The present invention is illustrated by way of example, and not

limitation, in the Figures of the accompanying drawings in which:

[0017]   **Figure 1** shows an example of a BLUETOOTH packet.

[0018]   **Figure 2** shows an example of a BLUETOOTH receive channel.

[0019]   **Figures 3a** through **3d** show a pair of correlation examples.

[0020]   **Figure 4a** shows an example of a series of binary values that

correspond to an 8:1 oversampling rate.

[0021]   **Figure 4b** shows an example of the series of binary values of Figure

4a after being decimated at a 2:1 decimation rate.

[0022]   **Figure 5a** shows an example of a series of binary values that

correspond to a 13:1 oversampling rate.

[0023]   **Figure 5b** shows an example of the series of binary values of **Figure

4b** after being decimated at a 3:1 decimation rate.

[0024]   **Figure 6a** shows a portion of a first correlation process if a received binary sample stream has a 544544... sampling pattern and the correlation word has a 544544... sampling pattern.

[0025]   **Figure 6b** shows a portion of a second correlation process if a received binary sample stream has a 454454... sampling pattern and the correlation word has a 544544... sampling pattern.

[0026]   **Figure 6c** shows a portion of a third correlation process if a received binary sample stream has a 445445... sampling pattern and the correlation word has a 544544... sampling pattern.

[0027]   **Figure 7a** shows a portion of a fourth correlation process if a received binary sample stream 701a has a 544544... sampling pattern and the correlation word 702a has a 54445444... sampling pattern.

[0028]   **Figure 7b** shows a portion of a fifth correlation process if a received binary sample stream 701b has a 454454... sampling pattern and the correlation word 702b has a 54445444... sampling pattern.

[0029]   **Figure 7c** shows a portion of a sixth correlation process if a received binary sample stream 701c has a 445445... sampling pattern and the correlation word 702c has a 54445444... sampling pattern.

8

[0030] **Figure 8** shows an embodiment of a BLUETOOTH channel that can perform decimation.

[0031] **Figure 9a** shows a portion of a fourth correlation process if a received binary sample stream 901a has a 544544... sampling pattern and the correlation word 902a has a 544454445... sampling pattern.

[0032] **Figure 9b** shows a portion of a fifth correlation process if a received binary sample stream 701b has a 454454... sampling pattern and the correlation word 902b has a 544454445... sampling pattern.

[0033] **Figure 9c** shows a portion of a sixth correlation process if a received binary sample stream 701c has a 445445... sampling pattern and the correlation word 902c has a 544454445... sampling pattern.

Description

Correlation

[0034]   Before engaging in a discussion as to how the correlation process can be made more efficient, a more detailed discussion of correlation is provided with respect to **Figures 3a** through **3d**. **Figure 3a** shows a portion of an exemplary received synchronization word 301 (which corresponds to a region of the slicer output signal 222 of **Figure 2**) and an correlation word 302a (which corresponds to correlation word 230 of **Figure 2**). A corresponding correlator output waveform 303a for these two words 301, 302a is provided in **Figure 3b**. From observation of **Figure 3a**, words 301 and 302a are clearly dissimilar. As a result, a low correlation peak 305a is observed in the correlator output waveform 303a of **Figure 3b**.

[0035]   **Figure 3c** shows another exemplary received synchronization word 304 and an exemplary correlation word 302b (which is the same as correlation word 302a shown in **Figure 3a**). A corresponding correlator output waveform 303b for these two codes 304, 302b is provided in **Figure 3d**. From observation of **Figure 3c**, codes 304 and 302b are clearly similar. As a result, a high correlation peak 305b is observed in the correlator output waveform 303b of Figure 3d.

[0036]   The differences between the pair of words 304 and 302b (e.g., differences 306, 307, 308 and 309) of **Figure 3c** are due to the non-ideal aspects of the slicer output signal 222, 224 shown back in **Figure 2**. That is, as the slicer unit 210 unit effectively provides an "approximation" of the transmitting device's original baseband signal, some margin of error between the received digital symbols 224 and the transmitter's original baseband signal are expected.

[0037]   Thus, if an "ideal" correlation word is used for the correlation described above (i.e., a representation having no errors resulting from the approximations and/or recovery attempts made by the A/D converter 201 or slicer unit 210 of **Figure 2**), a less than perfect correlation result is not only expected but also deemed acceptable in typical applications. The amount of allowable imperfection may vary from application to application, thus it is important to point out that the particular differences 206, 207, 208, and 209 of **Figure 2** are examples chosen for ease of illustration and understanding.

[0038]   A perfect correlation (i.e., when the compared signals are absolutely identical) results, in theory, with a correlator output waveform having a maximum amplitude, zero width spike. Such a spike 310a,b is superimposed on the correlator output waveforms of **Figures 3b** and **3d** for ease of understanding. The better the correlation between the two synchronization words the more the correlator output waveform begins to resemble a maximum amplitude, zero width

11

spike 310a,b. Thus note that the "better" correlation output waveform 303b of

Figure 3d has a spike having higher amplitude and narrower width (i.e., more

resembles spike 310b) than the "worse" correlation output waveform peak 305 of

Figure 3b.

[0039]   Typically, a peak threshold level (and/or pulse width) in the correlator

output waveform is used (in light of the tolerances and acceptable margins of

error in the slicer output waveform discussed above) to identify whether or not

the received code "matches" the code that is unique to the receiving device. An

exemplary peak threshold level 311a,b is shown in **Figures 3b** and **3d**.

[0040]   Because the peak 305a of the waveform in **Figure 3b** is below the

threshold 311a, the received word 301 of **Figure 3a** will be deemed as not

matching the correlation word 302a. Similarly, because the peak 305b of the

waveform in **Figure 3d** is above the threshold 311b, the received word 304 of

Figure 3c will be deemed as matching the correlation word 302b. As such, a

packet having the synchronization word 301 of **Figure 3a** will not be recognized

or formally received by the receiving device while a packet having the

synchronization word 304 of **Figure 3c** will be recognized and formally received

by the receiving device.

Inefficiencies of Correlation

[0041]   Referring briefly back to **Figure 2**, note that the slicer 210 output signal 222 represents a series of binary samples rather than a continuous waveform (such as digital symbol signal 224). Thus, the correlator 221 operates upon a series of discrete binary samples 222 rather than a continuous signal. In various embodiments, the operation of the correlator 221 may be conveniently described as akin to "counting the matches" that exist between the received series of binary samples from the slicer 210 and a series of discrete binary values that represent the correlation word 230.

[0042]   As such, the correlation word 230 of **Figure 2** may be maintained by the receiving device as a series of discrete binary values that outline the symbols of the "looked for" synchronization word. The correlator 221 correlates the binary values of the correlation word 230 with the binary samples of the received synchronization word. Accordingly, referring to **Figures 3b** and **3d**, the peak value 310a, 310b of the correlator output 303a, 303b corresponds to the number of binary samples in the received synchronization word that "match" the binary values of the correlation word. For example, if the A/D converter 201 of **Figure 2** produces 13 samples per received symbol (e.g., 13 samples for received symbol 211b, 13 samples for received symbol 212b, etc.) the slicer output signal 222 will

have 832 binary samples for a 64 bit synchronization word (13 samples/symbol x 64 symbols/sych word = 832 samples/synch word).

[0043] As such, by nature of the correlation process, storage space (e.g., register space, memory space or other means) for storing as much as 832 binary samples (e.g., for storing the correlation word 230) may be designed into (or otherwise made available to) the correlator unit 221. As large amounts of storage space correspond to inefficient design, a more efficient solution seeks to successfully perform the correlation process while using fewer binary samples from the received synchronization word.

Efficient Correlation Embodiment(s)

[0044] **Figure 4a** shows an example of a slicer output signal 422a the corresponding symbols being received. As seen in **Figure 4a**, a "101symbol sequence is observed where each symbol is sampled at an 8:1 oversampling rate. As such, the slicer output signal 422a has eight binary samples per received symbol. **Figure 4b** shows the slicer output signal 422a of **Figure 4a** after it has been decimated at a 2:1 rate.

[0045] Decimation is the act of reducing a number of samples; and, a decimation rate is the rate at which samples are reduced. Reducing the number of samples prior to correlation (e.g., by decimating the slicer output signal) effectively allows for an efficient implementation of a correlation process. That is,

14

as less samples are used in the correlation calculation, less storage resources are. This results in a more cost effective implementation of the correlation function.

[0046] A decimation rate of R:1 may be defined to signify the number of samples in the non decimated sample stream per sample in the decimated sample stream. Thus, for example, a 2:1 decimation rate corresponds to the elimination of every other sample; a 3:1 decimation rate corresponds to the elimination of every second and third sample; etc. An example of a 2:1 decimation rate, as alluded to above, is observed in Figure 4b.

[0047] **As Figure 4b** shows the exemplary slicer output signal 422a of **Figure 4a** after it has been decimated at a 2:1 rate, note that the decimated signal 422b of **Figure 4b** has one binary sample (each approximately located at times t_1, t_2, t_3, ...t_{12} as drawn in **Figure 4b**) per pair of binary samples within the non decimated signal 422a of Figure 4a. As the non decimated signal 422a of **Figure 4a** corresponds to an 8:1 oversampling rate (such that 8 binary samples exist per symbol in the received data stream), the decimated signal 422b provides 4 binary samples per received symbol.

[0048] With 4 binary samples per received symbol, for a 64 bit synchronization word, the correlation peak produced from a perfect correlation (e.g., where the received binary sample stream pattern perfectly matches the "looked for" binary

value pattern) corresponds to a magnitude of 256 bits (e.g., 4 x 64 = 256). As such, the size of the storage space used to perform the convolution process is conserved.

[0049]   An Oversampling rate of 8:1 can be used in BLUETOOTH applications because of the availability and ease of use of an 8.0 MHz reference crystal from which the oversampling clock (which is shown as oversampling clock 205 of **Figure 2**) is derived. In a sense, the combination of an 8:1 oversampling rate and a 2:1 decimation rate, as drawn in **Figures 4a** and **4b**, may be said to "work nicely" because the oversampling rate is an integer multiple of the decimation rate. That is 8/2 = 4 = integer.

[0050]   When the oversampling rate is an integer multiple of the decimation rate, the decimation activity results in a fixed number of binary samples per received data bit in the decimated signal 422b. As such, correlation can be performed in a straightforward manner because, at some point during the correlation process, the received synchronization word binary sample stream is aligned with binary values of the correlation word. Better said, when the number of binary samples per received symbols is fixed, a "looked for" synchronization word can be constructed (e.g., with the same, fixed number of binary values per symbol) that will allow for a 1:1 comparison of each received binary sample with each "looked for" binary value.

[0051] When the oversampling rate is not an integer multiple of the decimation rate, however, the automatic 1:1 alignment (during the correlation process) of each received binary sample with its corresponding "looked for" binary value is lost. **Figures 5a, 5b** and **6** and the comments that follow demonstrate this aspect in more detail.

[0052] **Figure 5a** shows another "101" symbol sequence where each bit of data is oversampled at a 13:1 oversampling rate. As such, the slicer output signal 522a has thirteen binary samples per received symbol. **Figure 5b** shows the slicer output signal 522a of **Figure 5a** after it has been decimated at a 3:1 rate. Note, therefore, that the decimated signal 522b of **Figure 5b** has one binary sample (each approximately located at times t_1, t_2, t_3, ...t_{13} as drawn in **Figure 4b**) for every three binary samples within the non decimated signal 522a of **Figure 5a**.

[0053] Oversampling rates of 13:1 are commonly used in BLUETOOTH applications because of the availability and ease of use of an 13.0 MHz reference crystal from which the oversampling clock (which, again, is shown as oversampling clock 205 of Figure 2) is derived. Note however, that a 13:1 oversampling rate and an 3:1 decimation rate does not result with the oversampling rate being an integer multiple of the decimation rate. That is, 13/3 = 4.333 = non-integer.

[0054]   As a result of the non-integer multiple relationship as between the

oversampling rate and the decimation rate, the decimation activity results in a

varied (rather than fixed) number of binary samples per received data bit in the

decimated signal 522b. Specifically, for example, note that the first received

symbol "1" of the "101" symbol pattern of **Figure 5b** has five binary samples (at

times t_1 through t_5) while the second and third received symbols "01" of the "101"

pattern each have four binary samples (at times t_6 through t_9 for received symbol

"0" and times t_{10} through t_{13} for received symbol "1").

[0055]   Thus, the "101" symbol pattern of **Figure 5b** may be said to have a

"544" sampling pattern because the first symbol consumes 5 samples, the

second symbol consumes 4 samples, the third symbol consumes 4 samples, etc.

An analysis of the situation will show that, the 544 sampling pattern repeats itself

over the course of any continued symbols. That is, a 5445445445... sampling

pattern emerges such that every third symbol has five samples and the

remaining symbols have 4 samples. The repeating sampling pattern is a natural

result of the non integer multiple relationship between the oversampling rate and

the decimation rate.

[0056]   The non integer relationship between the oversampling rate and the

decimation rate may be viewed as an integer plus a fraction. The integer portion

describes a "base" binary sample per received symbol rate and the fraction

describes a "fractional" binary sample per received symbol rate that is added to

18

the base rate. The base rate has a binary sample per received symbol rate of 1.0 or higher (depending on the specific combination of sampling rate and decimation rate) and the fractional binary sample per received symbol rate has a binary sample per symbol rate of less than 1.0 (again, depending on the specific combination of oversampling rate and decimation rate).

[0057]   For example, as described with respect to the example shown in **Figure 5b**, the oversampling rate divided by the decimation rate = 13/3 = 4+1/3 binary samples per received symbol. As such, the base rate corresponds to a rate of 4 binary samples per received symbol and the fractional rate corresponds to one binary sample per 3 received symbols (which is equal to 1/3 binary samples per received symbol). The 5445445445... sampling pattern may therefore be viewed as the base rate pattern 4444444444... plus the fractional rate pattern 1001001001... (i.e., 4444444444... + 1001001001... = 5445445445...)

[0058]   A complication, however, is that the arrival time of a BLUETOOTH packet is "random" in the sense that the exact phase position (or "starting point") of the first received symbol cannot be pre-determined to a high degree of accuracy. As such, the first symbol in a received data stream may be represented with 4 binary samples or 5 binary samples; and, as a result, it cannot be predetermined whether a 5445445445... sampling pattern applies, a 4454454454... sampling pattern applies, or a 4544544544... sampling pattern

19

applies (where the first value in each sampling pattern corresponds to the number of binary samples used to represent the first received symbol). Better said, because the starting point of the received symbols cannot be determined, neither can the starting point of the repeating sampling pattern.

[0059]   The inability to predetermine the exact arrival of the first symbol leads to correlation problems. Specifically, the same correlation word 230 will yield vastly different correlation peak values depending on the sampling pattern – even if the "looked for" synchronization word is being received. For example, as described in more detail below, a first sampling pattern can produce a maximum peak value (e.g., of 278 matched bits) while a second sampling pattern can produce a substantially lower maximum peak value (e.g., of 257 bits).

[0060]   **Figures 6a** through **6c** demonstrate this aspect in greater detail. **Figure 6a** shows a portion of the correlation process if the received binary sample stream 601a has a 544544... sampling pattern and the correlation word 602a has a 544544... sampling pattern. **Figure 6b** shows a portion of the correlation process if the received binary sample stream 601b has a 454454... sampling pattern and the correlation word 602b has a 544544... sampling pattern. **Figure 6c** shows a portion of the correlation process if the received binary sample stream 601c has a 445445... sampling pattern and the correlation word 602c has a 544544... sampling pattern.

20

[0061]   Note that the phase of the correlation word 602a,b,c sampling pattern

is the same for all three examples whereas the phase of the sampling pattern for

the received binary sample stream 601a,b,c varies from **Figure 6a** to **Figure 6b**

to **Figure 6c**. The examples of **Figure 6a** through **6c** illustrate the variation in

correlation "mismatches" that arise even if the same "looked for" synchronization

word is received. The correlation errors shown in **Figures 6a** through **6c** arise

solely from the differences in the number of samples for each received symbol;

thus, it may be assumed that all other samples are matched to their

corresponding correlation word value.

[0062]   In the example of **Figure 6a**, the correlation word 602a is phase

aligned with the received binary sample stream 601a. As such, no correlation

errors arise. With respect to the examples of **Figures 6b** and **6c**, however, the

correlation word 602b,c is not aligned to the received binary sample stream

601a,b. As such, under worse case conditions, 4 correlation errors per 6

received symbols (which is equal to 8 correlation errors per 12 received symbols)

arises in both of these cases. For a 64 bit synchronization word, this

corresponds to 43 "mismatches" out of 278 possible matches. That is, a

correlation peak of only 235 bits is obtained.

[0063]   Thus, even if the same synchronization word is being received, the

number of mismatches found during the correlation process can widely vary

(e.g., from 0 to 43 which corresponds to a correlation peak variation from 278 to

235) depending on the phase relationship between the sampling patterns of the received binary sample stream and the correlation word. Having such varied performance for the same received synchronization word results in uncertainty as to where the threshold level (e.g., threshold level 311a,b of **Figure 3**) should be placed.

[0064]   A mechanism for correlating a decimated received sample stream that reduces the ill effects of its phase being unknown involves the use of a "special" correlation word. Better said, a "special" correlation word may be constructed having a pattern that is designed to yield acceptably high correlation peak values for each of the different phases the received binary sample stream pattern may exhibit. As such, the "looked for" synchronization word can be identified upon reception regardless of its sampling phase. A discussion as to how a special correlation word may be constructed follows immediately below.

[0065]   Recall from the discussion above that a repeating sampling pattern is a natural result of the non integer multiple relationship between the oversampling rate and the decimation rate. That is, the integer portion describes a "base" binary sample per received symbol rate and the fraction describes a "fractional" binary sample per received symbol rate that is added to the base rate. As discussed above, for the example of **Figure 5b**, the oversampling rate divided by the decimation rate = 13/3 = 4+1/3 binary samples per received symbol. As such, the base rate corresponds to a rate of 4 binary samples per received

symbol and the fractional rate corresponds to one binary sample per 3 received symbols (which is equal to 1/3 binary samples per received symbol). The 5445445445... pattern may therefore be viewed as the base rate pattern 4444444444... plus the fractional rate pattern 1001001001... (i.e., 4444444444... + 1001001001... = 5445445445...).

[0066]   Note that the base rate may be viewed as not contributing to the correlation "mismatches" that arise as a result of the sampling phase. That is, as the base rate corresponds to a fixed number of binary samples per symbol (e.g., 4) the correlation process is immune to the phase of the received data stream sampling pattern with respect to its fixed rate (e.g., each symbol has at least 4 binary samples regardless of phase). As such, the correlation process may be viewed as being susceptible to the phase of the fractional rate.

[0067]   That is, for example, the three different received patterns discussed above (544..., 445..., 454...) may be viewed as being the same base rate sampling pattern (444...) with three different fractional rate sampling patterns (100..., 001..., 010...). As such, the "phase" of the binary sample stream (and the correlation problem(s) that arise as a result) may be viewed as growing out of the fractional portion of the sampling rate. Note that **Figures 6b** and **6c** are consistent with this perspective.

[0068] Therefore, an approach to developing a "special" correlation word (that correlates sufficiently well with a received stream of binary samples regardless of its sampling pattern phase) is to emphasize the base rate and de-emphasize the fractional rate. Better said, the fractional rate of the "special" correlation word can be made to be less than the fractional rate of the binary sample stream. For example, if the oversampling rate divided by the decimation rate is defined as N + k/n where N is the base rate (of binary samples per received symbol) and k/n is the fractional rate (of binary samples per received symbol), a special correlation word may be constructed where the binary values used for the correlation word correspond to a N + q/m sampling of the "looked for" synchronization word and where k/n > q/m.

[0069] For example, in one embodiment, the special correlation word is developed by forming a series of binary values that represent a 4+1/4 sampling of the "looked for" synchronization word. Note that, with respect to this particular embodiment, the correlation word will exhibit a 5444... repeating pattern rather than a 544... repeating pattern. For example, depending on the choice of phase for the special correlation word, various embodiments include the repeating patterns: 54445444...; 44454445...; or 44544454.... Note that any of these particular embodiment may be used to address the sampling and decimation approach of **Figure 5b**.

[0070]   As such, the summation of the base and fractional rates for the received binary sample stream corresponds to N+k/n = 4 + 1/3; and the summation of the base and fractional rates for the binary values of the correlation word correspond to N+ q/m = 4 + 1/4. Note that, in this case, N=4, k =q = 1, n=3 and m = 4. As such, (k/n = 1/3) > (q/m = 1/4). Thus, in summary, because correlation errors arise from phase mismatches and because phase mismatches arise from the fractional rate component of the oversampling rate to decimation rate ratio, better correlation will result with the special correlation word because its fractional rate component is less than the fractional rate component of the received samples.

[0071]   **Figures 7a** through **7c** demonstrate an embodiment of the special correlation word sampling pattern described above and, furthermore, may be viewed as an "apples to apples" comparison of the correlation improvement that results with respect to the example of **Figures 6a** through **6b**. **Figure 7a** shows a portion of the correlation process if the received binary sample stream 701a has a 544544... sampling pattern and the correlation word 702a has a 54445444... sampling pattern. **Figure 7b** shows a portion of the correlation process if the received binary sample stream 701b has a 454454... sampling pattern and the correlation word 602b has a 54445444... sampling pattern. **Figure 7c** shows a portion of the correlation process if the received binary sample stream 701c has a 445445... sampling pattern and the correlation word 702c has a 54445444... sampling pattern.

[0072] Note that, again, the phase of the correlation word 702a,b,c sampling pattern is the same for all three examples in **Figures 7a,b,c** whereas the phase of the sampling pattern for the received binary sample stream 701a,b,c varies from **Figure 7a** to **Figure 7b** to **Figure 7c**. The examples of **Figure 7a** through **7c** illustrate the improvement in correlation "mismatches" that arise if the same "looked for" synchronization word is received. The correlation errors shown in **Figures 7a** through **7c** arise solely from the differences in the number of samples for each received symbol; thus, it may be assumed that all other samples are matched to their corresponding correlation word value.

[0073] As seen in each of **Figures 7a** through **7c**, 5 mismatches arise per 12 received symbols. Thus, there is more consistency and less mismatches across the different received data stream sampling phases if the special correlation word is employed. With respect to consistency, note that a mismatch rate of 5 mismatches per 12 received symbols is observed for all three sampling pattern phases of **Figures 7a,b,c** rather than a variation between 0 and 8 mismatches per 12 received samples as observed in **Figures 6a,b,c**.

[0074] With respect to less mismatches, 5 mismatches per 12 received symbols is less than 8 mismatches per 12 received symbols. Consistent with this view, note that 5 mismatches per 12 received symbols will produce 26 or 27 mismatches for a 64 bit synchronization word. Thus, for 278 total binary

26

samples, a correlation peak of 251 or 252 matches can be achieved (which is an improvement over the 235 matches of **Figures 6b** and **6c**). Thus, a higher correlation peak threshold may be applied without concern as to the phase of the received sample stream sampling pattern.

[0075]   Note that, with respect to **Figures 7a** through **7c**, each of the received sample streams 701a,b,c have a higher overall sample per symbol ratio than the correlation word. That is, received sample streams 701a,b,c each have 52 binary samples per 12 symbols ((12x4)+4=48+4=52) whereas the correlation words 702a,b,c each have 51 binary samples per 12 symbols ((12x4)+3=48+3=51). Thus, the correlator may be designed to correlate two signals that possess a different sampling rate from each other.

[0076]   In alternate embodiments, the correlator may be designed to handle identical sampling rates for the pair of signals to be correlated. Various approaches may be taken. For example, according to one approach, the correlation word is "bit stuffed" so that its overall sampling rate is increased to that of the received binary sample stream. For example, referring to **Figures 7a** through **7c**, the correlation words 702a,b,c may each be modified to include one more sample per 12 symbols into each of the correlation words 702a,b,c. This will raise the sampling rate of the correlation word from 51 samples per 12 symbols to 52 samples per 12 samples.

[0077] As just one example of this approach, the first symbol in the correlation word (e.g., symbol 703a, 703b, 703c) may be represented with 6 rather than 5 samples. Alternatively, any other symbol may have its sample representation increased. This can produce an additional mismatch per 12 symbols (e.g., from 5 mismatches per 12 symbols to 6 mismatches per 12 symbols) which will add 5 more mismatches to a 64 bit synchronization word. As such, in an embodiment, the correlation peak can be 247 or 248 (rather than 251 or 252). Nevertheless, note that improvements are still realized.

[0078] If the correlation word is bit stuffed so that its sampling rate is equal to that of the received sample stream, it will have an overall fractional sampling rate that is equal to the overall fractional sampling rate of the received sample stream. For example, as discussed with respect to **Figures 6a** through **6c** and **7a** through **7c**, the sampling rate of the received sample stream 601a,b,c, 701a,b,c is 4+1/3. This corresponds to a fractional rate of 1/3 samples per symbol.

[0079] Furthermore, as discussed, the sampling rate of the correlation word 702a,b,c of **Figures 7a** through **7c** is 4+1/4. By stuffing an additional sample per 12 symbols into the correlation word, the sampling rate of the correlation rises to 4+1/3. However, the correlation word may be not only be viewed as having an overall fractional rate of 1/3 but also as having a first and second fractional rates (1/4 and 1/12) that add up to the fractional rate of the received sample stream (1/4). That is, 1/4 + 1/12 = 4/12 = 1/3. The first fractional rate component 1/4,

28

corresponds to the reduced fractional rate q/m discussed above; and, the second fractional rate component 1/12 corresponds to the difference between the fractional rate of the received sample stream and reduced fractional rate q/m.

[0080]   Alternative to bit stuffing the correlation word, the received sample stream may be decimated a second time in order to drop the sampling rate of the received data stream to that of the correlation word. For example, in the embodiment of **Figures 7a** through **7c**, the received data streams 701a,b,c may be decimated at a rate of 1 sample per 12 symbols. This corresponds to the removal of one sample from each of the received data streams 701a,b,c observed in **Figures 7a** through **7c** which drops their corresponding sampling rate from 52 samples per 12 symbols to 51 samples per 12 samples.

[0081]   Similar to the bit stuffing approach, decimating the received sampling stream in this manner can produce an additional mismatch per 12 symbols (e.g., from 5 mismatches per 12 symbols to 6 mismatches per 12 symbols) which will add 5 more mismatches to a 64 bit synchronization word. As such, again, in an embodiment the correlation peak can be 247 or 248 (rather than 251 or 252).

[0082]   In even further alternate embodiments, bit stuffing the correlation word and a second decimating of the received sample stream may be combined together (rather than just one or the other) so that the correlation operates on input signals having the same rate. For example, the received sample stream

may be decimated at a rate of 1 sample per 24 symbols and the correlation word may be stuffed at a rate of 1 sample per 24 symbols. This corresponds to a sampling rate for both the received sample stream and the correlation word of 51.5 samples per 24 symbols.

[0083]   **Figure 8** shows an embodiment of a channel design 809 that includes a decimator 850 between the slicer 810 and the correlator 821. The decimator 850 decimates the slicer output signal 822 in accordance with the appropriate decimation. The decimator 850 may be implemented, for example, with logic that performs the decimation function or with a processor that executes a software routine configured to perform the decimation function. If two decimations (or more) are to be performed, more decimators can be added between the decimator 850 and the correlator 821. The appropriate "special" correlation word 830 can be stored (e.g., in register or memory space) and made accessible to the correlator 821.

[0084]   Note that the teachings above allow for a 1:2 decimation rate (as discussed with respect to **Figures 4a** and **4b**) and a 1:3 decimation rate (as discussed with respect to **Figures 7a** through **7c**) to be alternatively available in the same receiving channel. That is, for example, the receiving channel 809 of **Figure 8** may be designed to have two modes: an 8 MHz mode and a 13 MHz mode. During the 8 MHz mode the oversampling clock 805 frequency is derived from an 8 MHz clock and the correlation is performed in accordance with the

teachings of **Figures 4a** and **4b** (i.e., a 2:1 decimation rate is performed by decimator 850).

[0085]   During the 13 MHz mode the oversampling clock 805 frequency is derived from a 13 MHz clock and the correlation is performed in accordance with the teachings of **Figures 7a** through **7c** (i.e., a 3:1 decimation rate is performed by decimator 850 and a special correlation word is employed). It is important to note that those of ordinary skill will be able to apply the teachings above to applications other than 8 MHz based oversamplings, 13 MHz based oversamplings, BLUETOOTH applications, 2:1 decimation rates, 3:1 decimation rates, 544... sampling patterns, 5444... special correlation word sampling patterns, etc. That is, those of ordinary skill will be able to configure working correlation approaches for their particular application, where possible, that are different than the specific approaches discussed above.

[0086]   **Figures 9a** through **9c** relate to another embodiment of a "special" correlation word that does not require any bit stuffing (or a second decimation to the received sample stream). As discussed, because the phase position of the received sample stream cannot be predetermined, a number of different received sample stream patterns are possible (e.g., a 544... received sampling pattern, a 454... received sampling pattern, or a 445... received sampling pattern). Therefore a "special" correlation word having different components, where each component is representative of a different possible received sample stream

31

pattern, will yield acceptably high correlation peak values for each of the different phases the received binary sample stream pattern may exhibit. As such, the "looked for" synchronization word can be identified upon reception regardless of its sampling phase.

[0087] For example, as discussed for an oversampling rate to decimation rate ratio of 13:3, a 544... received sampling pattern, a 454... received sampling pattern, or a 445... received sampling pattern are each possible. Therefore, constructing a "special" correlation word having a representative component of each of these patterns (e.g., a 544 454 445 ... pattern) will yield a sufficiently high correlation peak for any of the received sampling pattern phases. That is, because each possible received sampling pattern phase is "represented" in the correlation word, a sufficiently strong correlation with the "looked for" synchronization word will result for any of the received sampling pattern phase possibilities.

[0088] Furthermore, in a further refinement, by representing each received sampling pattern phase equally within the correlation word, small variation in correlation peak value should be observed across the different received sampling stream patterns. **Figures 9a** through **9c** show this aspect in more detail. **Figures 9a** through **9c** show the number of possible errors that may result if a 544 454 445 ... correlation word 902a,b,c is correlated against a received sampling stream having a 544... sampling pattern 901a (as observed in **Figure**

9a), a 454... sampling pattern 901b (as observed in **Figure 9b**) and a 445 ... sampling pattern 901c (as observed in **Figure 9c**).

[0089]   As a 544 454 445 ... sampling pattern repeats itself every 9 bits, two repeating patterns worth of samples (i.e., 18 symbols worth of samples) are shown in **Figures 9a** through **9c**. As observed in **Figures 9a** through **9c**, 8 errors per 18 symbols may result for each of the three types of received sampling pattern phases. For a 64 bit synchronization word, this can produce a total of 28 or 29 errors. Out of 278 possible "matches", this corresponds to a correlation peak of 249 or 250.

[0090]   Note that a more careful scrutiny of exactly how errors occur during the correlation will show that the actual number of errors in **Figure 9a** could be as low as 6 errors per 18 symbols (which would produce a correlation peak of 257); the actual number of errors in **Figure 9b** could be as low as 4 errors per 18 symbols (which would produce a correlation peak of 264); and the actual number of errors in **Figure 9a** could be as low as 6 errors per 18 symbols (which would produce a correlation peak of 257). Regardless, correlation results that compare to an approach that de-emphasizes the fractional rate (as described above with respect to **Figures 7a** through **7c**) are achieved. Alternate embodiments would include repeating sampling patterns of: 445 454 544 ...; 454 544 445...; 445 544 454...; 544 445 454...; or 454 445 544...

[0091] Note also that embodiments of the present description may be implemented not only within a semiconductor chip but also within machine readable media. For example, the designs discussed above may be stored upon and/or embedded within machine readable media associated with a design tool used for designing semiconductor devices. Examples include a netlist formatted in the VHSIC Hardware Description Language (VHDL) language, Verilog language or SPICE language. Some netlist examples include: a behaviorial level netlist, a register transfer level (RTL) netlist, a gate level netlist and a transistor level netlist. Machine readable media also include media having layout information such as a GDS-II file. Furthermore, netlist files or other machine readable media for semiconductor chip design may be used in a simulation environment to perform the methods of the teachings described above.

[0092] Thus, it is also to be understood that embodiments of this invention may be used as or to support a software program executed upon some form of processing core (such as the CPU of a computer) or otherwise implemented or realized upon or within a machine readable medium. A machine readable medium includes any mechanism for storing or transmitting information in a form readable by a machine (e.g., a computer). For example, a machine readable medium includes read only memory (ROM); random access memory (RAM); magnetic disk storage media; optical storage media; flash memory devices; electrical, optical, acoustical or other form of propagated signals (e.g., carrier waves, infrared signals, digital signals, etc.); etc.

[0093]   In the foregoing specification, the invention has been described with reference to specific exemplary embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention as set forth in the appended claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.